THE UNIVERSITY OF TEXAS AT DALLAS

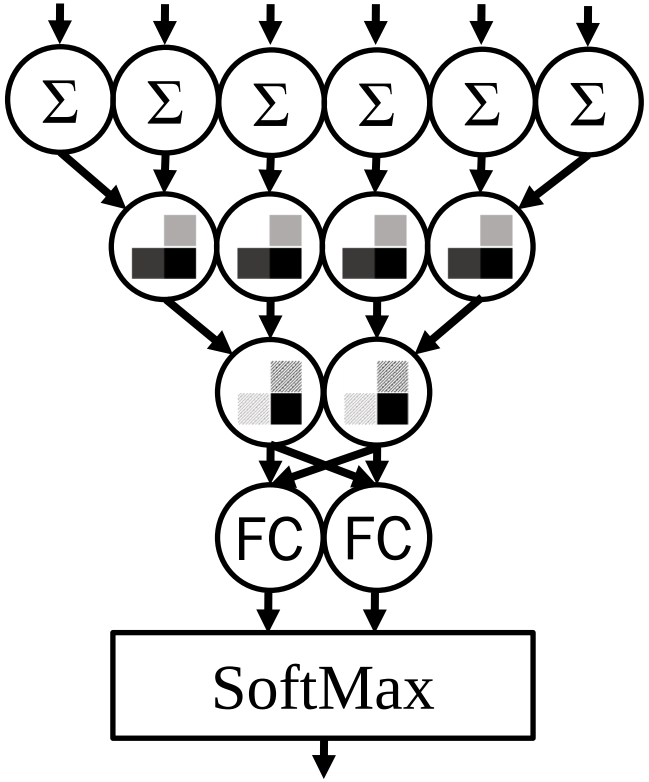
NeuOS: A Latency-Predictable Multi-Dimensional Optimization Framework for DNN-driven Autonomous Systems

Soroush Bateni
The University of Texas at Dallas

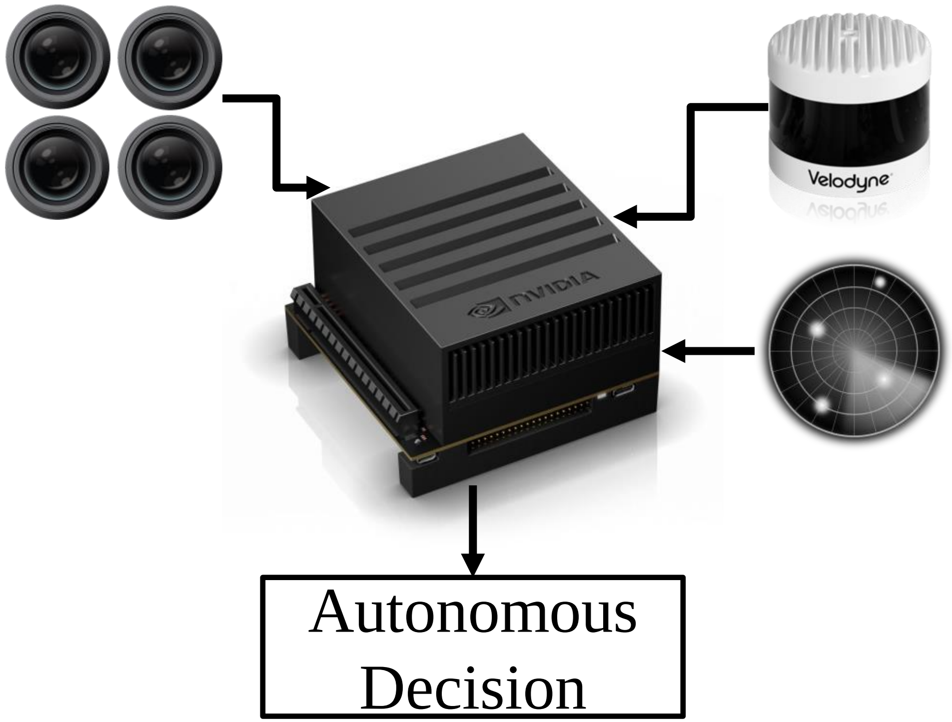
Cong Liu
The University of Texas at Dallas

The tale of two worlds

Deep Neural Networks (DNNs)

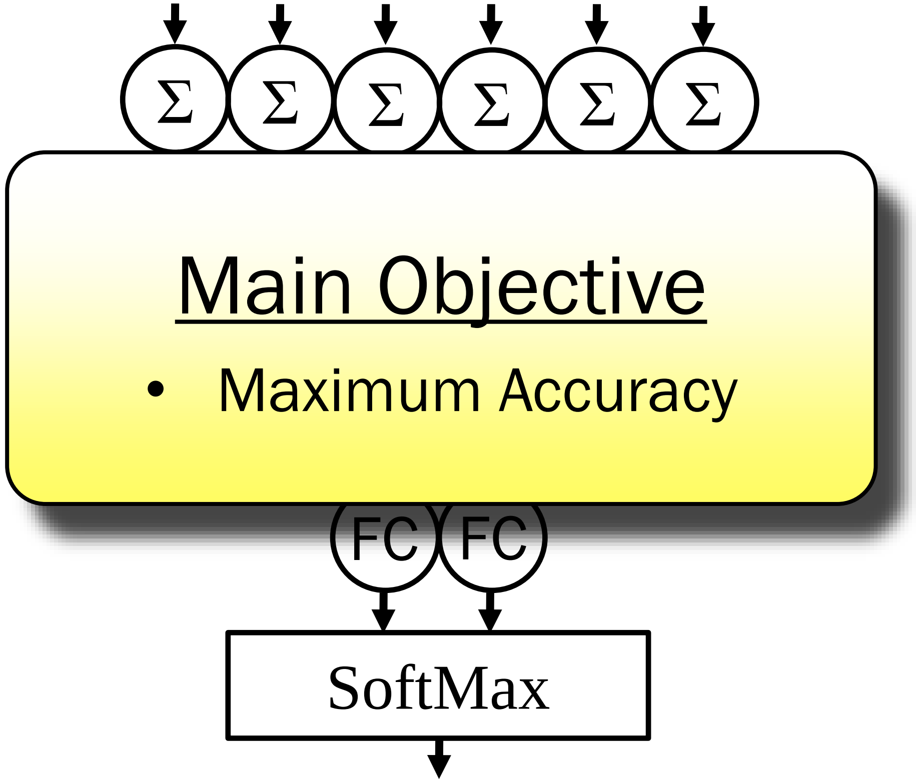


Autonomous Embedded Systems

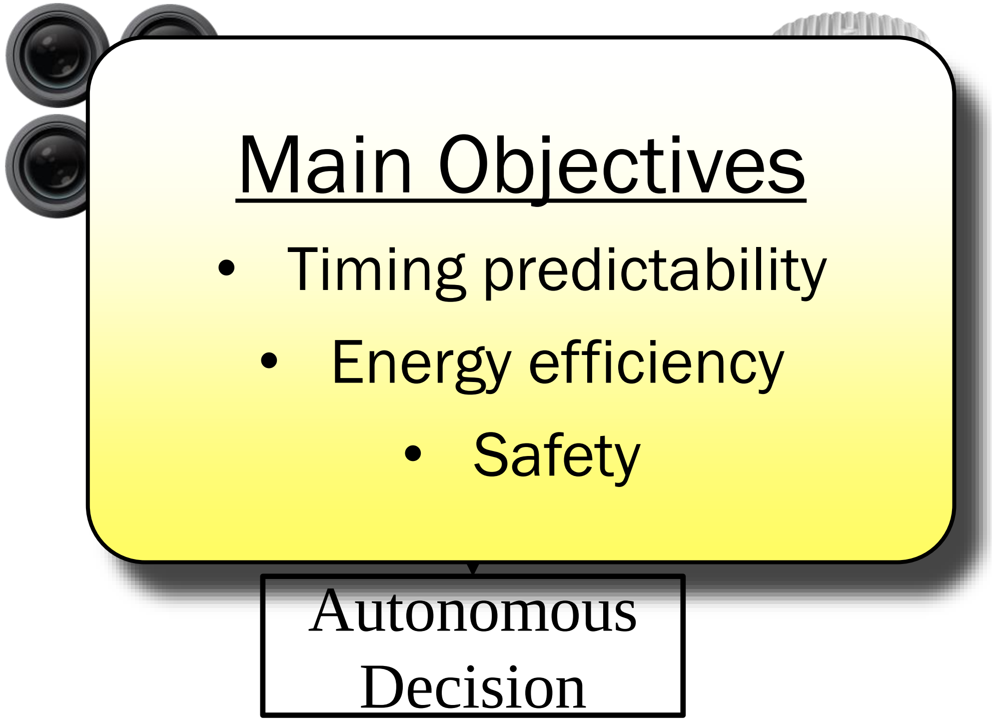


The tale of two worlds

Deep Neural Networks (DNNs)



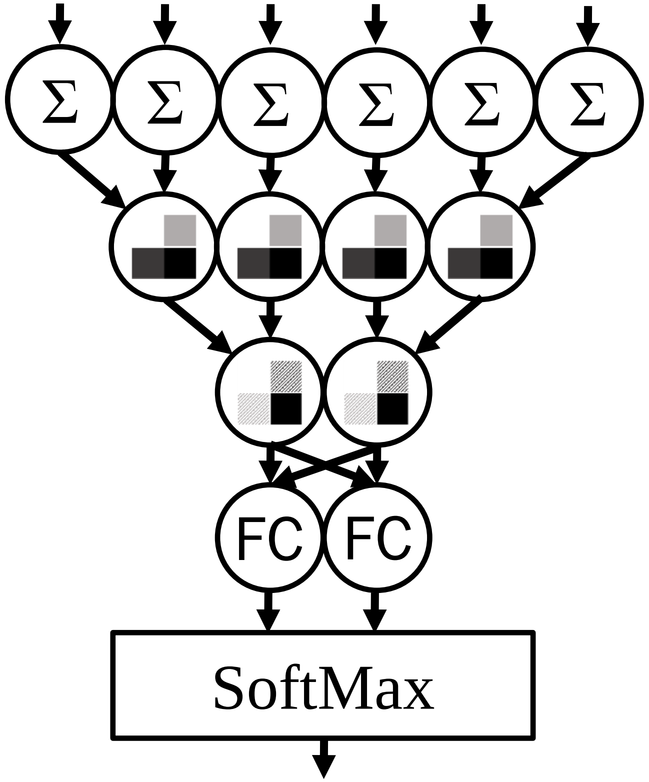
Autonomous Embedded Systems

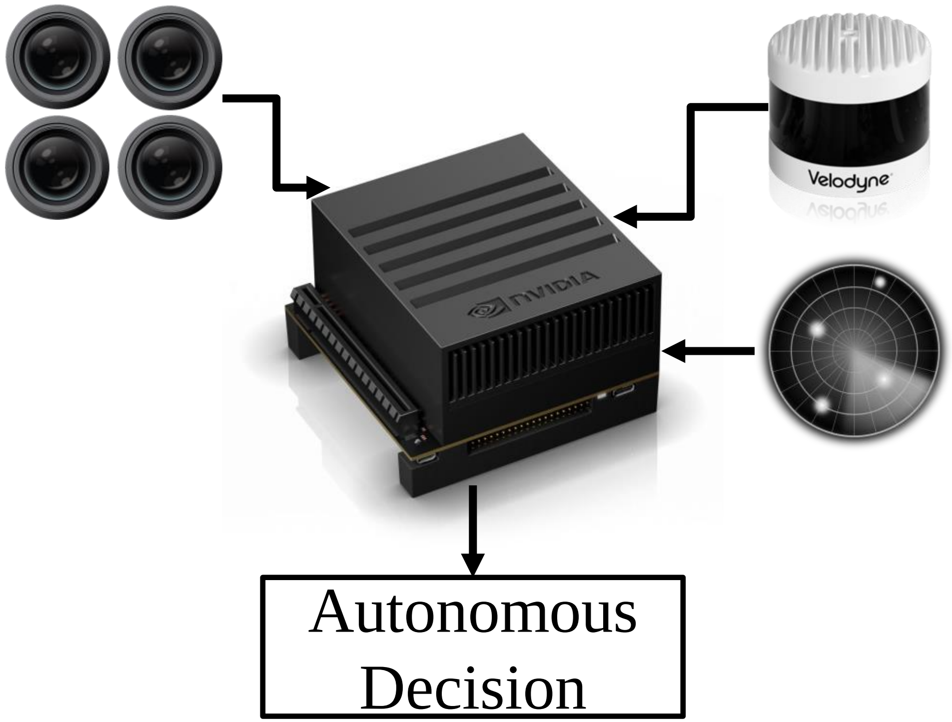


Marriage between the two worlds

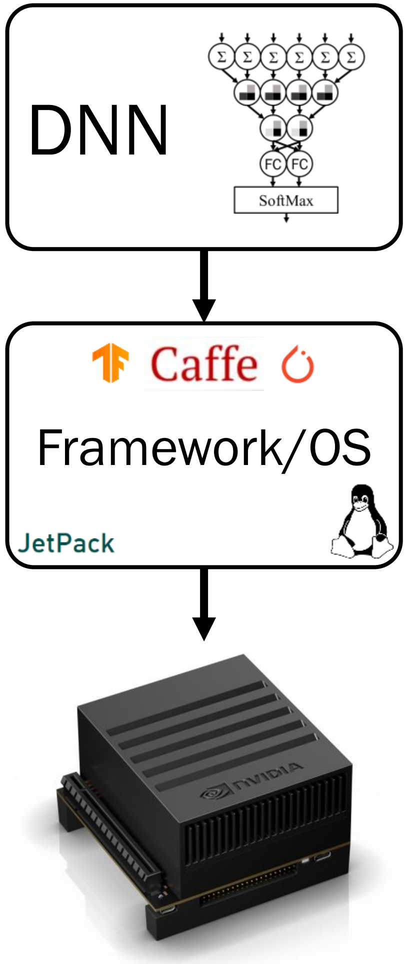
Deep Neural Networks (DNNs)

Autonomous Embedded Systems



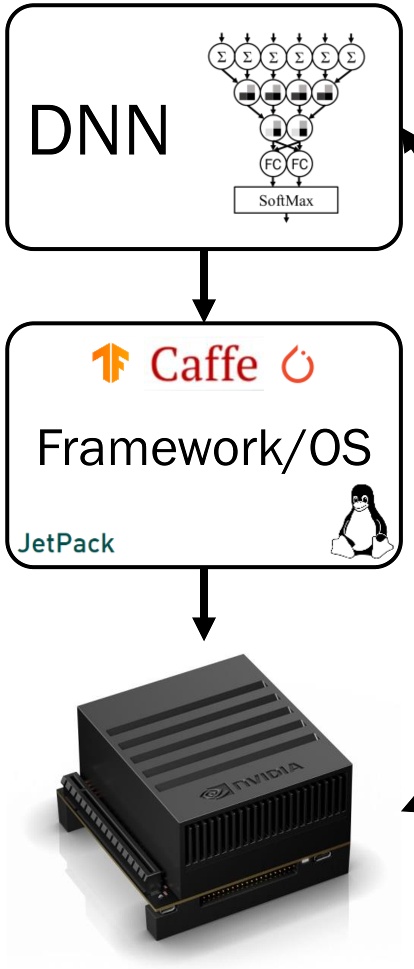


The big picture



Hardware/software stack for executing DNNs in Autonomous Embedded Systems

The big picture

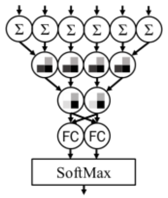


Hardware/software stack for executing DNNs in Autonomous Embedded Systems

The focus of related research in AES is currently mostly on the DNN and the hardware.

The big picture

DNN



Caffe

Framework/OS

JetPack

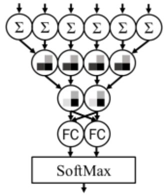
Efficient DNNs

- Quantization
- Lowrank approximation

Embedded Systems

The big picture

DNN



Caffe

Framework/OS

JetPack

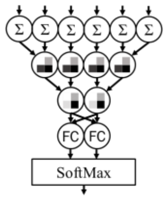
Hardware/software stack for executing DNNs in Autonomous Embedded Systems

Special Processors

- AI accelerators
- DNN-focused SoCs

Where system software/framework can help

DNN



Caffe

Framework/OS

JetPack

Challenges

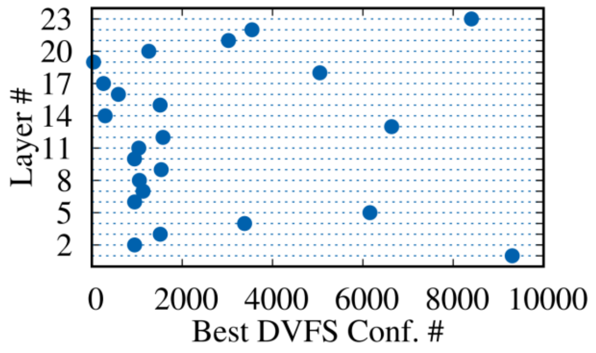
- Meet timing requirements
- Be energy efficient
- Minimize accuracy loss.

All the above goals must be achieved at the same time.

Jack of all trades, master of none

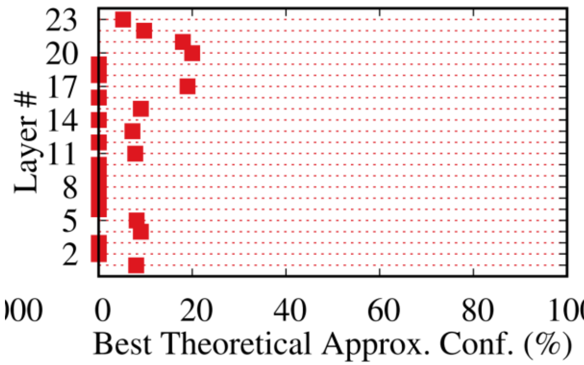
Timing predictable & energy efficient

Can be achieved at system level via Dynamic Voltage Frequency Scaling (DVFS).



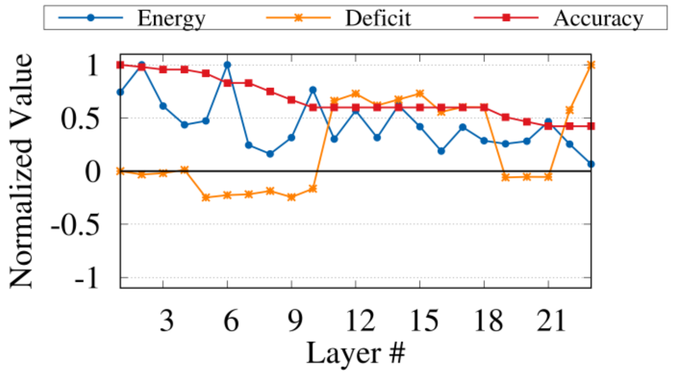
Timing predictable & accurate

Can be achieved at application level via DNN configuration change.



Master of none

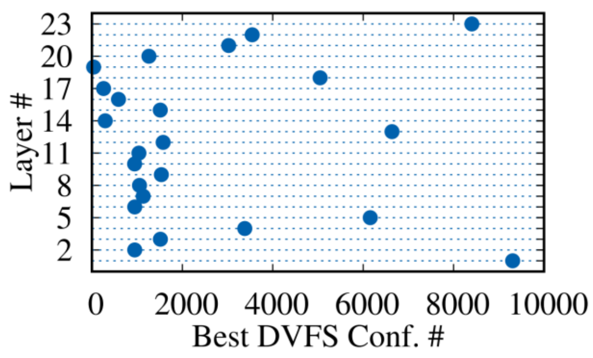
Combining the two (even at different rates) will yield unpredictable results.



Jack of all trades, master of none

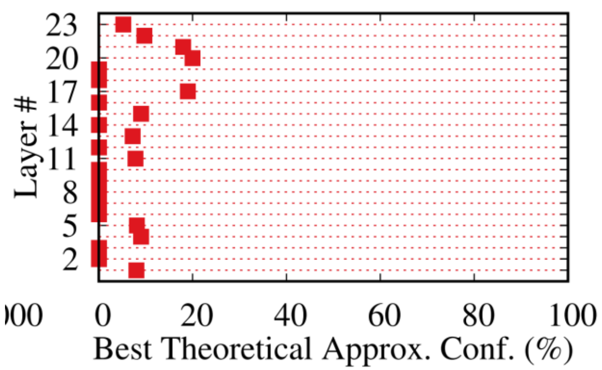
Timing predictable & energy efficient

Can be achieved at system level via Dynamic Voltage Frequency Scaling (DVFS).



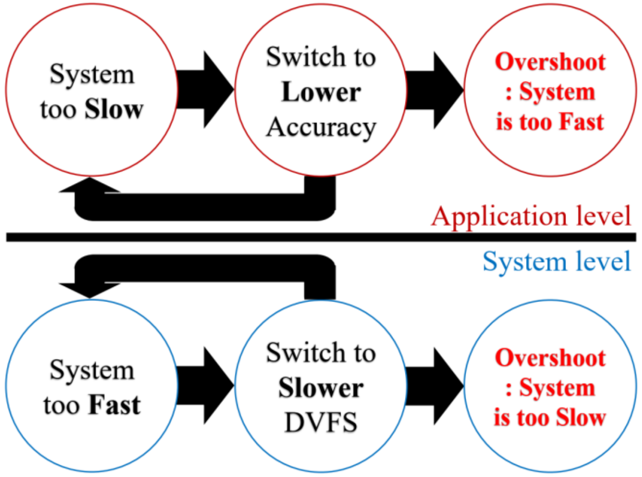
Timing predictable & accurate

Can be achieved at application level via DNN configuration change.



Master of none

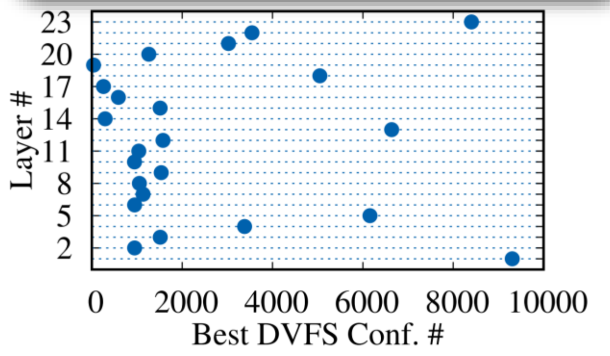
Combining the two (even at different rates) will yield unpredictable results.



Jack of all trades, master of none

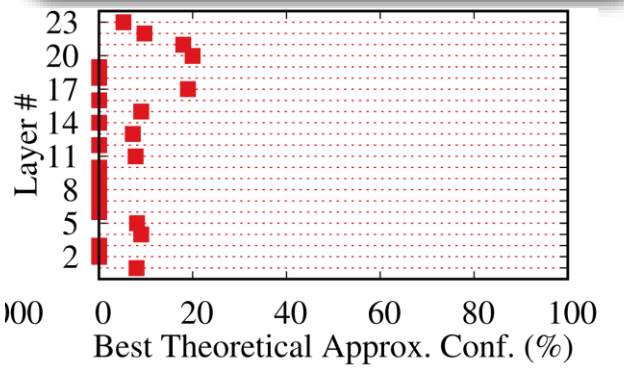
Timing predictable & energy efficient

Need per-layer adjustments.



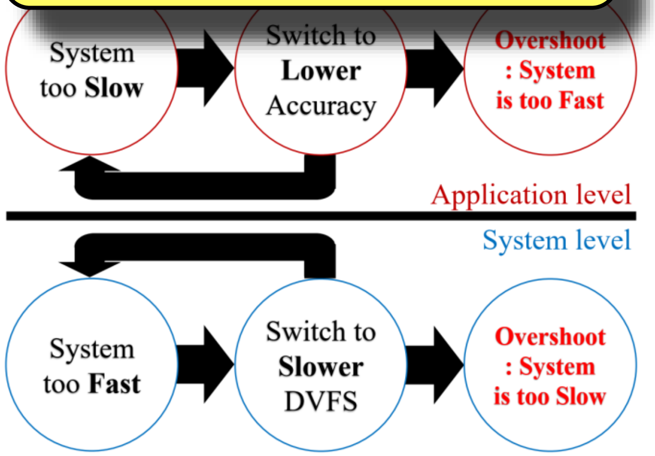
Timing predictable & accurate

Need per-layer adjustments.



Master of none

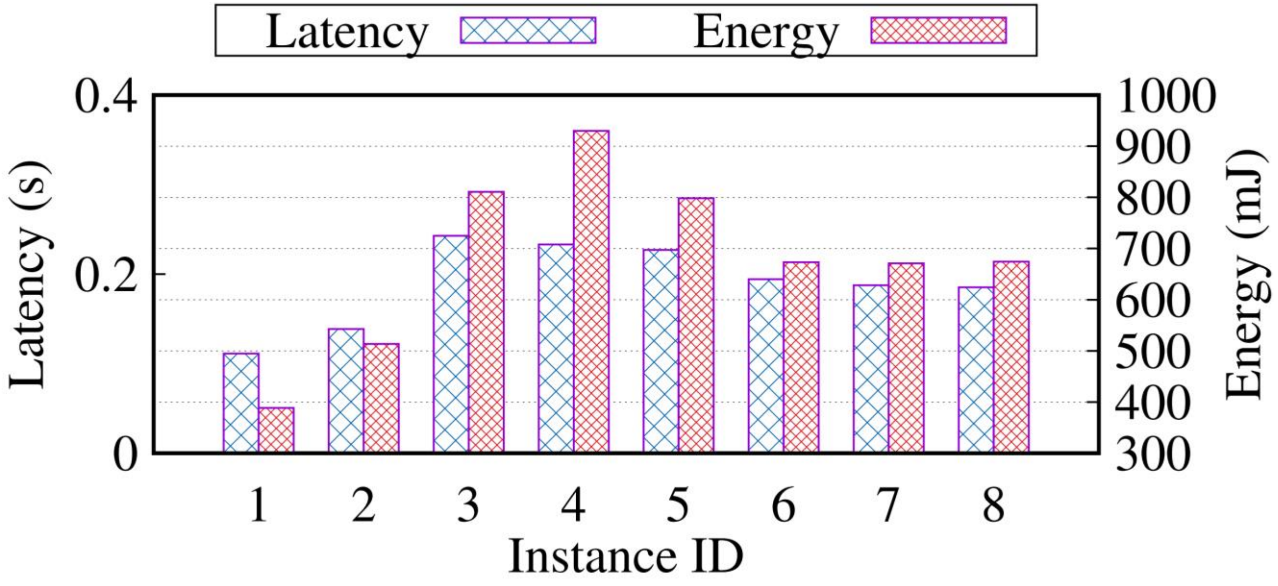
Need coordination.



No one is alone

Multiple ResNet-50 instances executed together

The underlying system-level solution here is PredJoule[1]



Takeaways

The first DNN instance is the winner, other DNN instances not as lucky because the method used here is greedy.

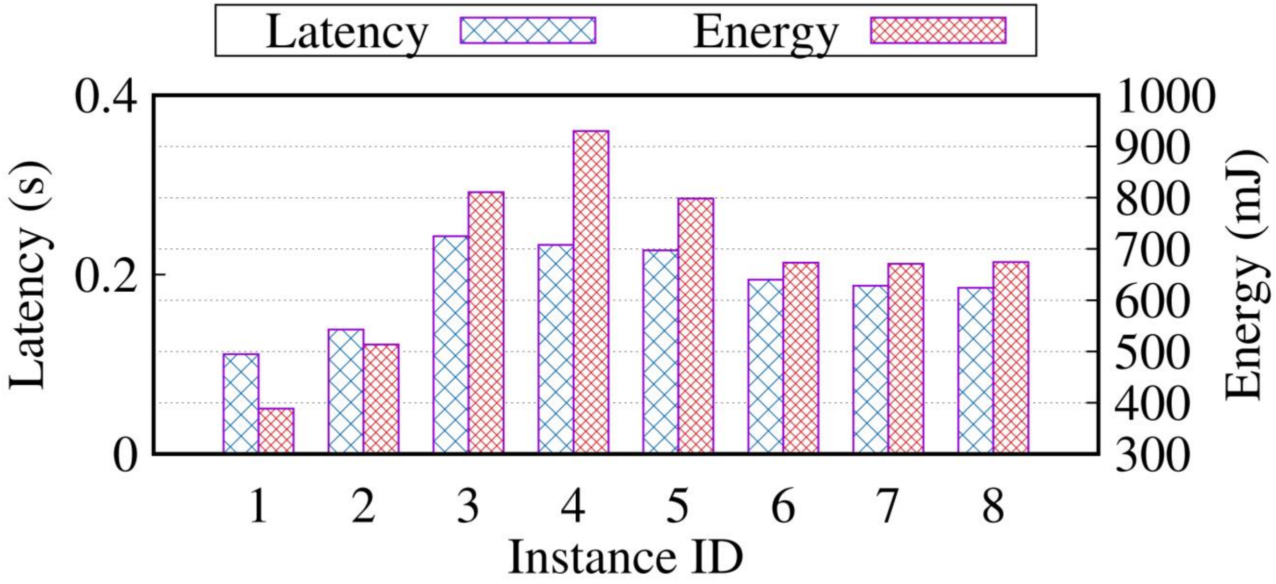
The DVFS configurations chosen only work well for the first DNN instance.

[1]Bateni, Soroush, Husheng Zhou, Yuankun Zhu, and Cong Liu. "Predjoule: A timing-predictable energy optimization framework for deep neural networks." In *2018 IEEE Real-Time Systems Symposium (RTSS)*

No one is alone

Multiple ResNet-50 instances executed together

The underlying system-level solution here is PredJoule[1]



Takeaways

The first DNN instance is
the winner, other DNN

Need cross-DNN coordination.

The DVFS configurations chosen only work well for the first DNN instance.

[1]Bateni, Soroush, Husheng Zhou, Yuankun Zhu, and Cong Liu. "Predjoule: A timing-predictable energy optimization framework for deep neural networks." In *2018 IEEE Real-Time Systems Symposium (RTSS)*

Design Goals

Core Targets

- **Timing predictable**: the system must meet deadlines set by the system designer for the DNN.
- **Energy efficient:** the system must use DVFS to achieve near-optimal energy usage for DNNs.
- **Accurate**: the system can change accuracy dynamically but must do so cautiously.
- **Multi-DNN compatibility**: the system should be able to coordinate and find an efficient solution for all DNN instances.

Optimization Targets

The system must also be flexible to adapt to different system constraints. We offer three optimization targets (switchable by an external policy controller):

- **Min Energy (M_p)** is used when our design is deployed in extremely low power scenario such as remote sensing.
- **Max Accuracy (M_A)** is used when our design is deployed in extremely mission-critical scenarios.
- **Balanced Energy and Accuracy** is the scenario where our design can choose what is best given the timing requirement.

Timing predictability

LAG analysis

- Keep track of per-layer progress

Tracked execution time

$$LAG_i(t, L_i(t)) = \sum_{l \in L_i(t)} (d_l - e_l)$$

Accumulative LAG

Per-layer sub-deadline

Proportional Deadline

- Build an ideal schedule by setting sub deadlines in proportion to their execution time

Per-layer execution time

$$d_l = (e_l / \sum_{x \in L_i} (e_x)) \cdot D_i$$

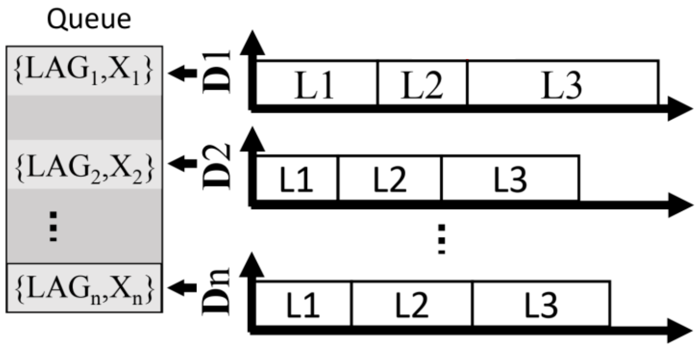
Per-layer sub-deadline

End-to-end deadline for the DNN instance

Coordination

Building a cohort

We keep a pair of local variables for each DNN instance.



Δ Calculator

1. Based on the last reported values of LAG in the cohort, calculate a speedup (or slowdown)
2. Lookup[1] the best possible DVFS configuration for that slowdown.
3. The output is a list (Δ) of optimal DVFS configurations for each DNN instance.

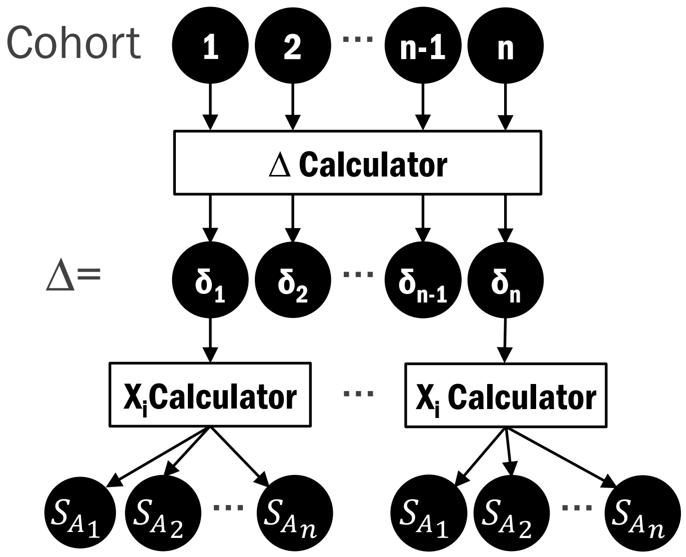
X_i Calculator

1. For each element of Δ, calculate the required (further) speedup (or slowdown) for other DNN instances.
2. This time, lookup[1] the best possible approximation configuration that matches that slowdown.

[1]Please see the paper and the source code for more information.

Optimization

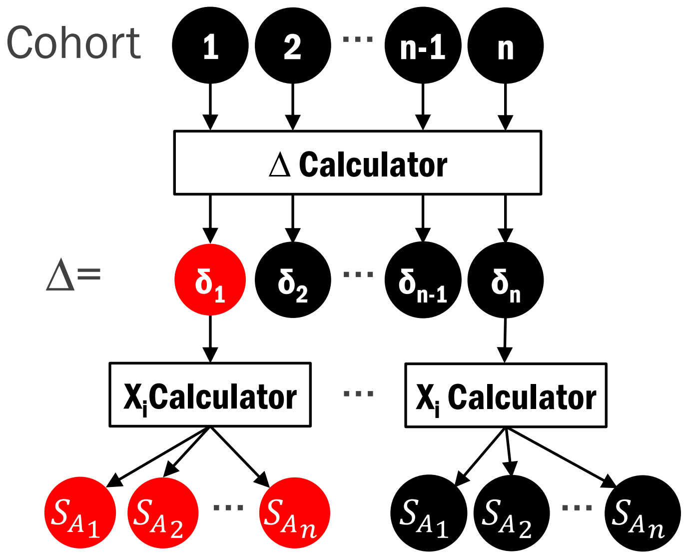
The decision tree



Overview of modes

Optimization

The decision tree



Overview of modes

Choosing a δ (DVFS configuration) will have consequences in terms of accuracy for all DNNs in the cohort. Therefore, the question is, which δ is the best?

Min. Energy (M_p) chooses the δ that has the least PowerUp value in the PowerUp/SpeedUp table, without looking at accuracy loss.

Max. Accuracy (M_A) chooses the δ so to minimize the value of $\sum_{\forall \delta_i} S_{A_i}$.

Balanced Energy and Accuracy uses the Bivariate Regression Analysis (BRA) to achieve a balanced approach backed by statistical analysis of the tree[1].

[1]Please see the paper for more information.

Overview

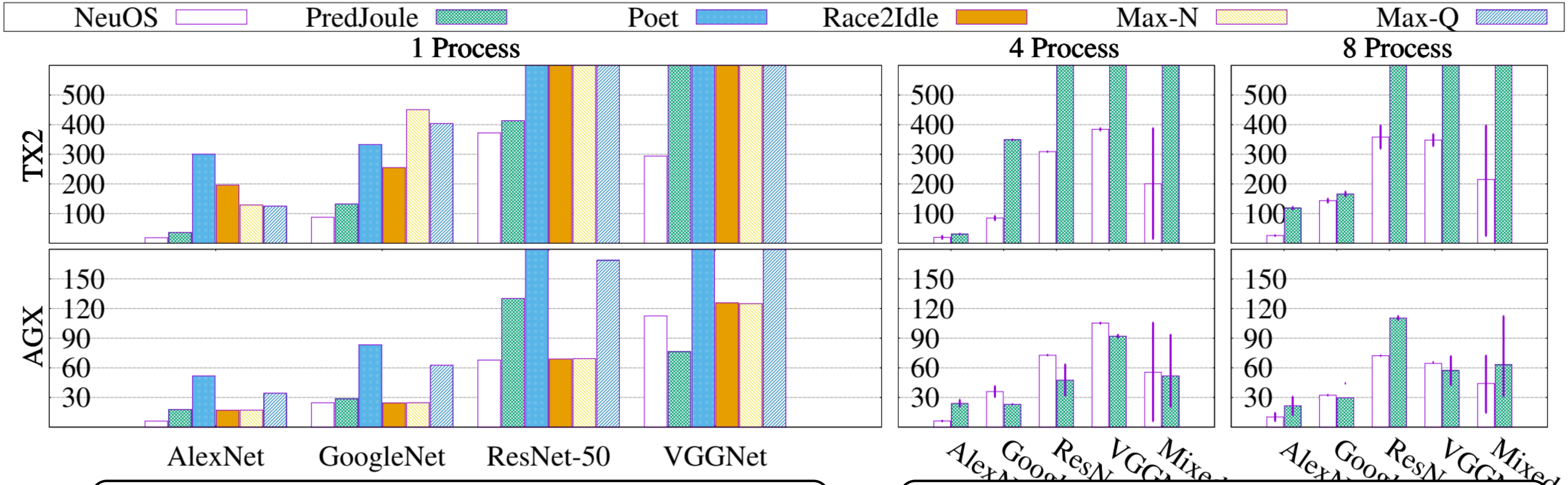
Based on Caffe

- Available as an open-source project on GitHub
- No need to use APIs
- No need to redesign DNN models
- Need to generate
  - Hash tables
  - Lowrank approximated version of your DNN model.

Tested extensively

- Tested on NVIDIA Jetson TX2 and Jetson AGX Xavier
- Tested using image recognition DNNs
  - AlexNet, GoogleNet, ResNet-50, VGGNet
- Tested using three cohort sizes
  - Small: 1 DNN instance
  - Medium: 2-4 DNN instances
  - Large: 6-8 DNN instances
- We include a mixed scenario that uses a combination of all the DNN models

Energy

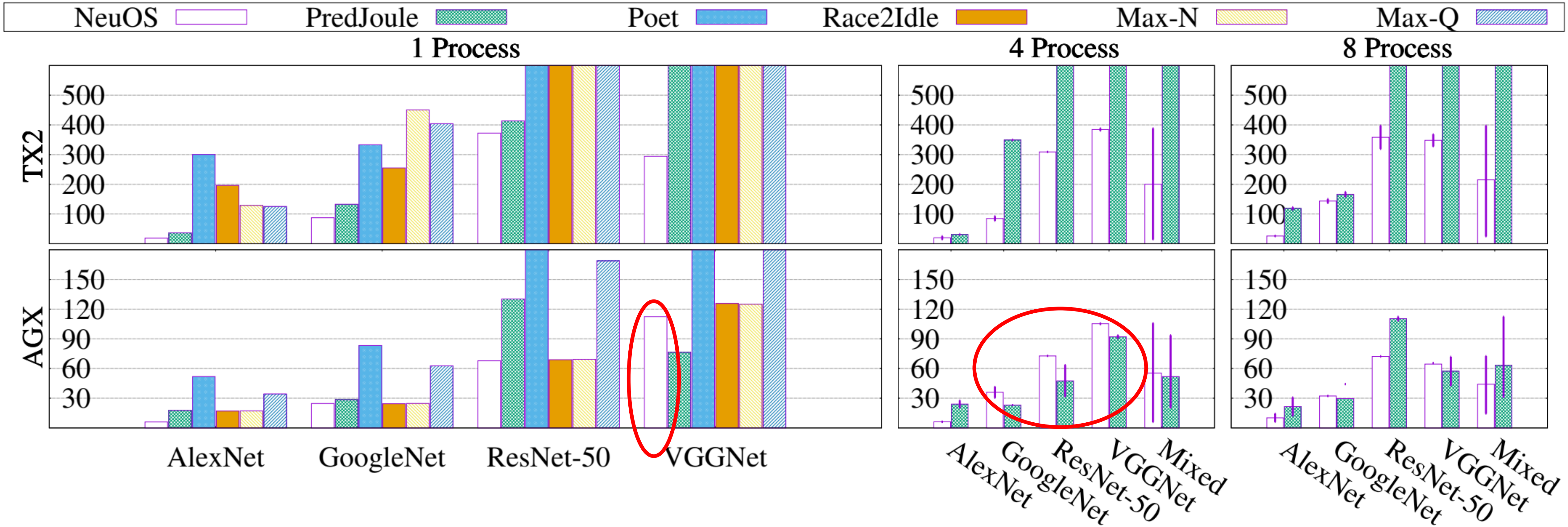


68% avg. improvement on TX2

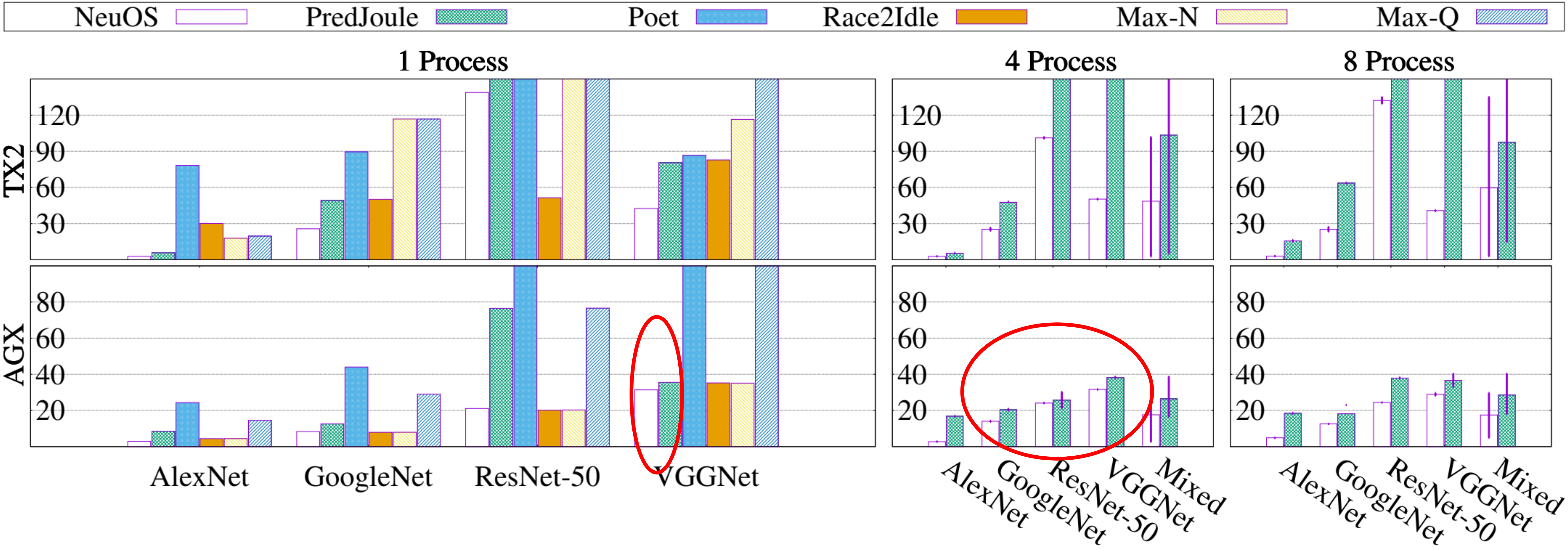
46% avg. improvement on AGX Xavier

70% avg. improvement on TX2

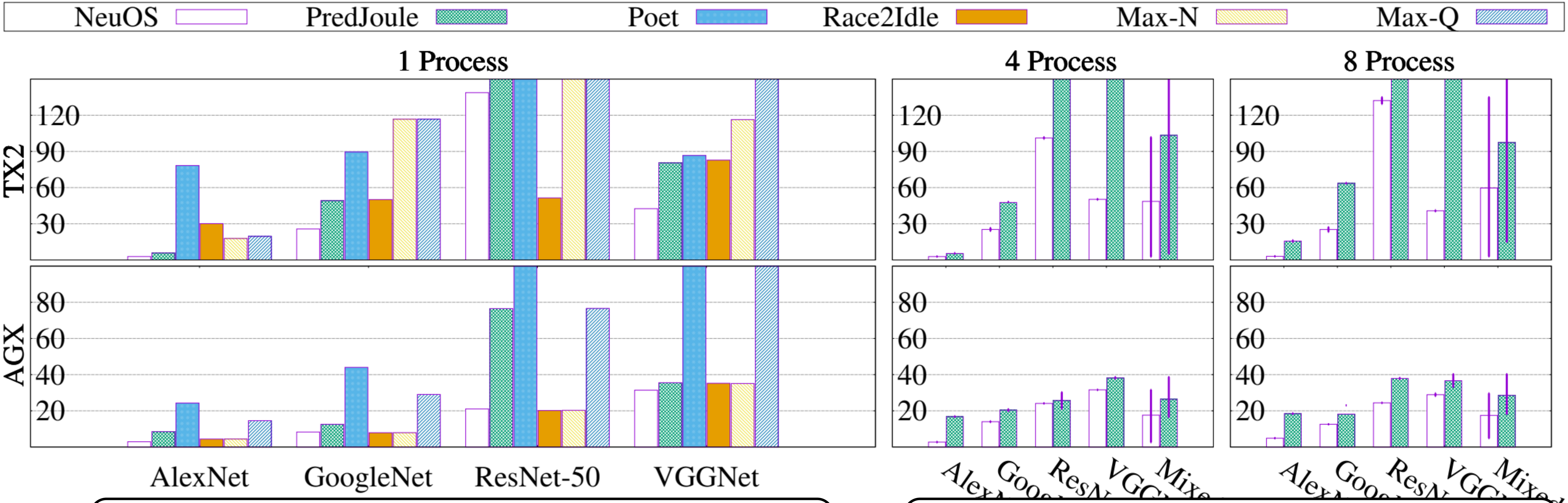
Energy


NeuOS
PredJoule
Poet
Race2Idle
Max-N
Max-Q
1 Process
4 Process
8 Process
TX2
AGX
500
400
300
200
100
150
120
90
60
30
AlexNet
GoogleNet
ResNet-50
VGGNet
Mixed

Latency


NeuOS
PredJoule
Poet
Race2Idle
Max-N
Max-Q
1 Process
4 Process
8 Process
TX2
AGX
120
90
60
30
80
60
40
20
AlexNet
GoogleNet
ResNet-50
VGGNet
Mixed

Latency



68% avg. improvement on TX2

40% avg. improvement on AGX Xavier

53% avg. improvement on TX2

32% avg. improvement on AGX Xavier

Tail Latency

Small cohort

3.25% deadline miss rate.

	TX2	AGX Xavier
AlexNet	9.2ms	5ms
GoogleNet	48ms	12ms
ResNet-50	130.3ms	26.1ms
VGGNet	39.1ms	36.2ms

Medium cohort

Deadline miss rate same as the small cohort.

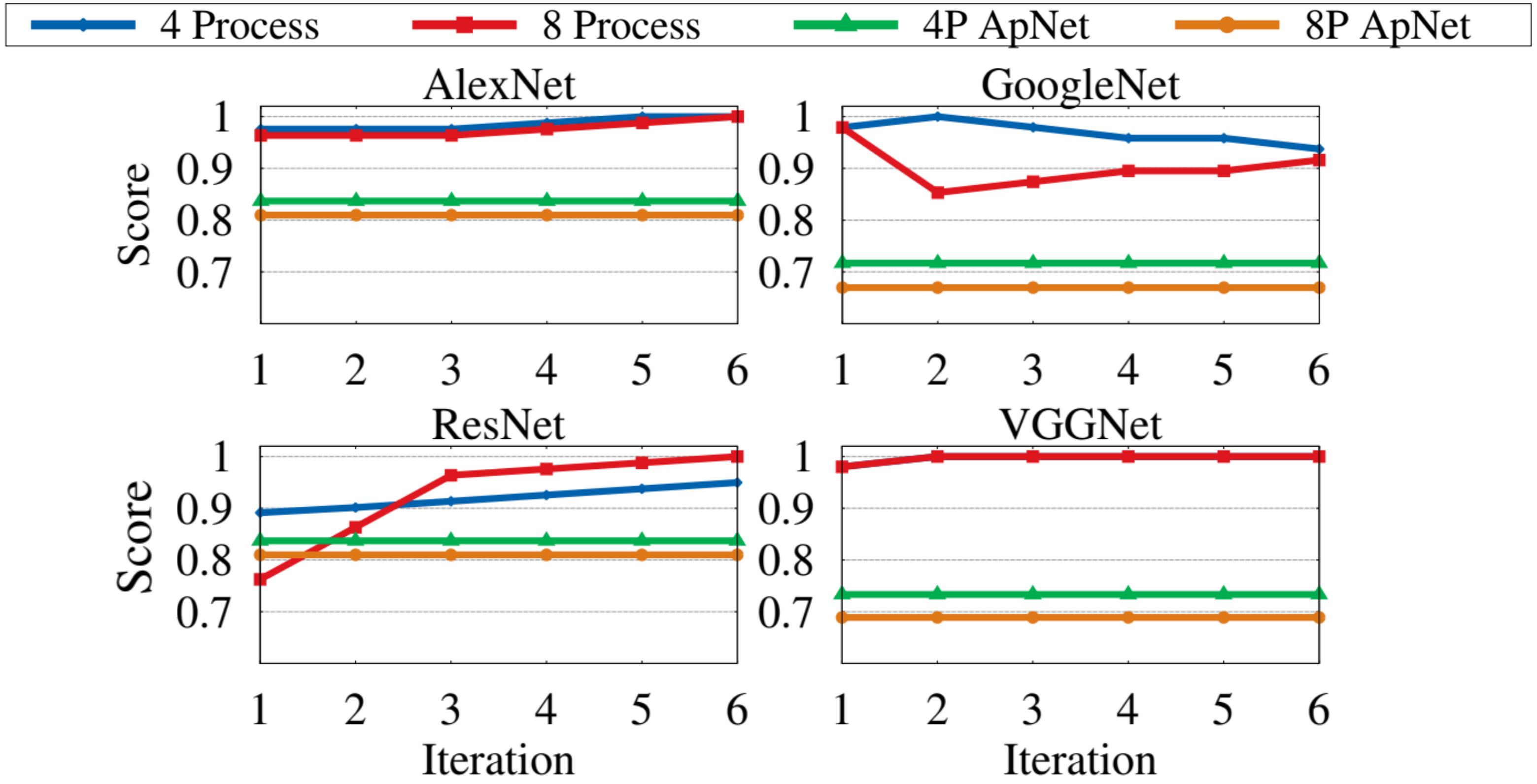
	TX2	AGX Xavier
AlexNet	10.4ms	11ms
GoogleNet	39.2ms	12.5ms
ResNet-50	101.7ms	26.3ms
VGGNet	69ms	25.9ms

Large cohort

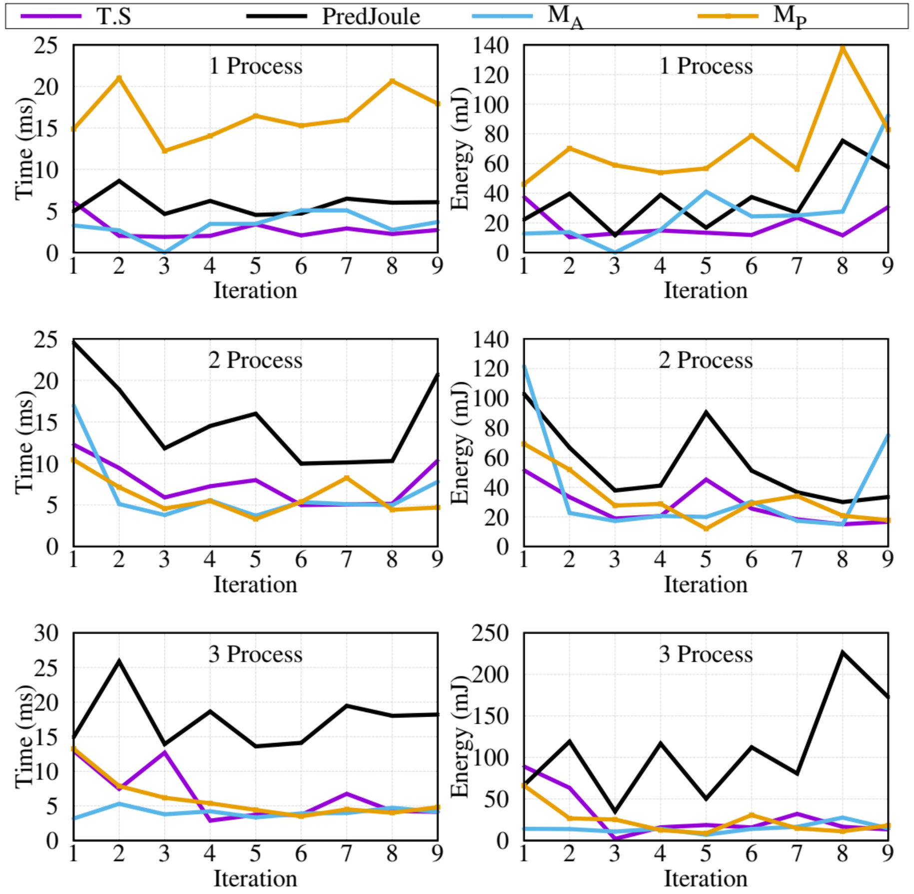
Deadline miss rate same as the small cohort.

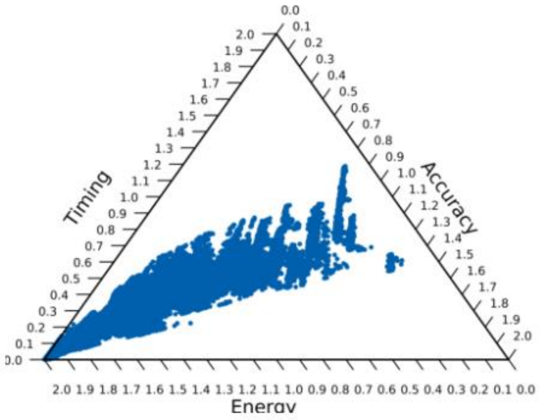
	TX2	AGX Xavier
AlexNet	13.6ms	10.7ms
GoogleNet	40.8ms	54ms
ResNet-50	190ms	62ms
VGGNet	72ms	36.1ms

Relative Accuracy

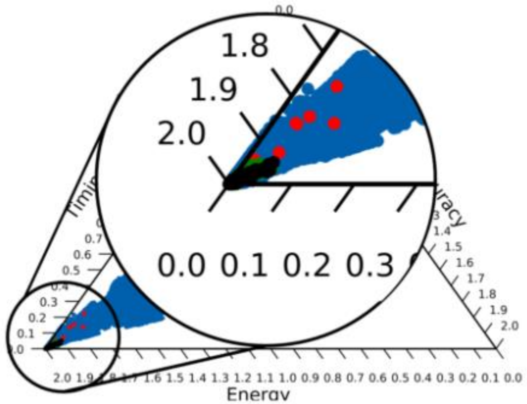


Flexibility



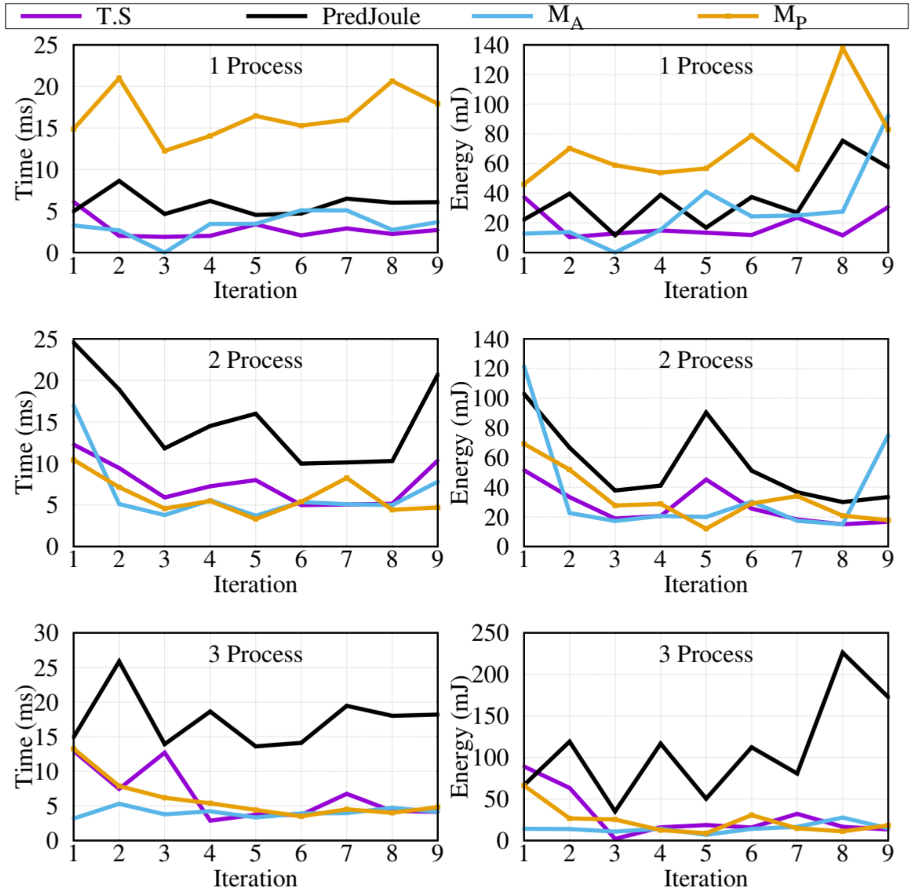


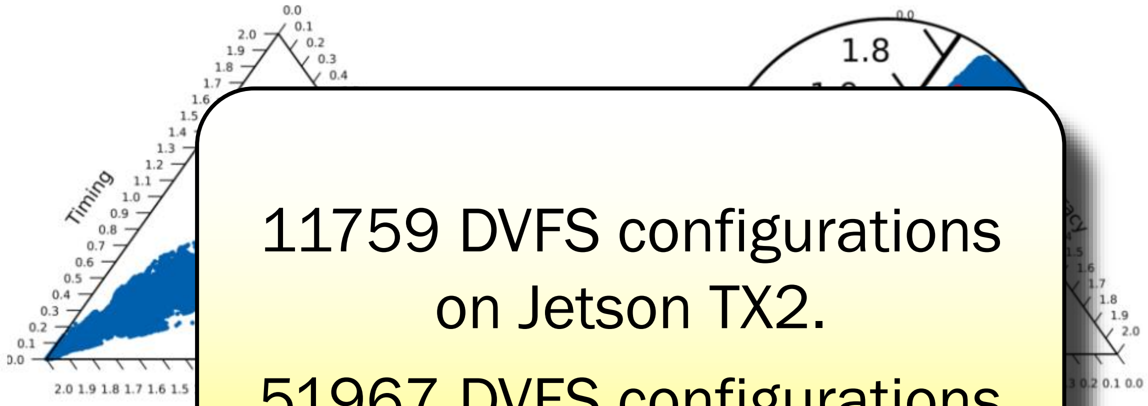
(a) The entire configuration space for all DVFS and accuracy combinations for Jetson TX2.



(b) Chosen configurations in the triangle space.

Flexibility





(a) The enti
for all DVFS
tions for Jets
in the

11759 DVFS configurations on Jetson TX2.

51967 DVFS configurations on Jetson AGX Xavier.

Overhead

Computation

Relatively negligible execution overhead (in ms).

	1 Process	4 Process	8 Process
NeuOS	0.145	0.571	0.738
PredJoule	0.772	0.929	1.597
ApNet	0	3.27	5.85
Poet	151.03	604.12	1208.27

Memory

Overhead includes the lowrank version of each DNN model. The right side shows how much of the total memory of each device is occupied.

	Overhead in addition to Caffe			Ratio to total memory	
	Lowrank	Hash Table	Ratio	Jetson TX2	AGX Xavier
AlexNet	226 MB	331 B	49%	10%	4%
GoogleNet	23 MB	2.1 KB	30%	2%	1%
ResNet-50	82 MB	3.2 KB	45%	7%	3%
VGGNet	509 MB	634 B	48%	25%	12.7%

The system community to the rescue

- Certain problems cannot be solved at application level (by AI researchers) and at hardware level separately
  - Ensuring timing predictability, energy efficiency, and accuracy for DNNs in Autonomous Embedded Systems requires coordination
- We presented the design of NeuOS that can achieve these three goals by
  - Using LAG analysis to ensure real-time performance
  - Efficiently propagating all possible choices
  - Having flexibility in terms of choosing the best combination of configurations based on system designer’s criteria or external policy controller
- We extensively evaluated NeuOS
  - Using the latest AES devices
  - Using prominent image recognition DNNs
  - Under multiple configurations, including various cohort sizes
  - Against the most prominent accessible solutions available to researchers.

Thank you

Questions

Please do not hesitate to send your questions to *soroush@utdallas.edu*.

Source Code

https://github.com/Soroosh129/NeuOS